Vortex Preservation Using a Coupled Eulerian-Lagrangian Solver

S. J. Huntley*, D. P. Jones† and A. L. Gaitonde‡
University of Bristol, Bristol, BS8 1TR, United Kingdom

This paper presents a coupled Vortex Particle Method-Computational Fluid Dynamics solver. The Vortex Particle Method is used to prevent dissipation of the vortex structure on coarse CFD meshes. Implementation of the approach uses the Split Velocity Method that specifies the fluid velocity as the sum of the induced vortex particle velocity and a remaining velocity. Dissipation of the vortex velocities on coarse meshes is removed and the CFD equations solved for the remaining velocity have an identical form to those for a moving mesh, but with additional source terms. The coupled solver is demonstrated on a selection of two-dimensional test cases and the results are compared to the solutions of the CFD solver on its own using a coarse mesh and a fine mesh. It is shown that the coupled solver preserves the vortices on a coarse mesh and is computationally more efficient than using the fine mesh.

Nomenclature

E	=	Energy
Pr	=	Prandtl number
p	=	pressure
Re	=	Reynolds number
t	=	time
u, v, w	=	total velocity components
$\hat{u}, \hat{v}, \hat{w}$	=	CFD particle-induced velocity components
$\hat{u}_p, \hat{v}_p, \hat{w}_p$	=	VPM particle-induced velocity components
$\tilde{u}, \tilde{v}, \tilde{w}$	=	background velocity components
Γ	=	particle strength
γ	=	ratio of specific heats
μ	=	dynamic viscosity
ρ	=	density

*Research Associate, Department of Aerospace Engineering, University of Bristol, Bristol, BS8 1TR, United Kingdom
†Reader in Aerodynamics, Department of Aerospace Engineering, University of Bristol, Bristol, BS8 1TR, United Kingdom
‡Reader in Aerodynamics, Department of Aerospace Engineering, University of Bristol, Bristol, BS8 1TR, United Kingdom

I. Introduction

UNDERSTANDING the effect of concentrated vortices interacting with a lifting surface is vital in many aspects of aerospace engineering. Examples of this include helicopter blade-vortex interaction and aircraft take-off and landing. Blade-vortex interaction occurs when the trailing wake from a helicopter rotor interacts with the oncoming blade. This causes unsteady loading on the blades and aerodynamic noise [1]. Wingtip vortices that are generated on aircraft take-off and landing determine separation distances to ensure the safety of following aircraft. The ability to accurately predict these types of flows represents a challenge in fluid simulation methods. The literature shows that Lagrangian, Eulerian and hybrid Eulerian/Lagrangian Computational Fluid Dynamics (CFD) techniques have been applied to these kinds of flows.

Eulerian CFD methods are widely used for aerospace applications due to their ability to accurately and efficiently resolve the flow near solid boundaries. However, they suffer from diffusive behaviour meaning their application to the simulation of flows with highly concentrated vortical regions has been limited, as accurately resolving these flow features would require a very fine mesh throughout the computational domain and result in a very large computational overhead.

Eulerian CFD methods used to compute vortical flows range in complexity and computational cost from Reynolds-Averaged Navier-Stokes (RANS) codes to Direct Numerical Simulation (DNS) methods. The RANS approach was used in Potsdam *et al.* [2] to calculate helicopter blade-vortex interactions. It is the least complex of the CFD methods but can become computationally expensive as it requires a very fine mesh to prevent excessive diffusion. This is highlighted in the study by Abate [3], where RANS was used to study an airfoil-vortex interaction in two dimensions, however the extension of this to three dimensions is unlikely due to the cost of using a fine mesh. More complicated than RANS is Large Eddy Simulation (LES), which has been used to simulate vortical flows in both two [4, 5] and three dimensions [6]. DNS is the most accurate of the CFD methods as it resolves all the turbulent length scales and does not model anything. However, it is computationally very expensive and so far has only been applied to simple aerospace flows such as a stalled NACA 0012 aerofoil [7]. Other studies have utilised high-order algorithms to increase the accuracy of the CFD in high gradient regions such as used by Svard *et al.* [8] to study the interaction of a vortex with a NACA 0012 airfoil. An alternative that has been explored is to adopt a chimera grid approach that uses overlapped grids to transport the vortex on a locally fine mesh across a coarser background mesh as investigated by Wolf [9]. It is also possible to use fully-coupled fluid-structure-interaction schemes to investigate flows with vortical regions as done by Malan and Oxtoby [10].

Lagrangian methods benefit from the fact that they do not suffer from any diffusive effects and computational effort can be reduced as computational elements only need to be placed in regions of interest. However these elements are isotropic in nature causing difficulty when modelling solid boundaries [11]. Lagrangian methods have repeatedly been used over the past few decades to model aircraft wakes and tip vortices. Rossow [12] used a vortex filament method to

model a simplified wake structure whilst Smith and Kroo [13] used a vortex panel method for the same purpose. More recently, Chatelain *et al.* [14] used a vortex particle method with a billion particles to model the aircraft wake. Vortex Lattice methods are the most widely used method for rotor wake prediction to represent the shed and trailed vorticity generated by the rotor, see for example Röttgermann *et al.* [15] and the study by Padakannaya [16]. Vortex Particle Methods have also been used to predict the rotor wake and blade-vortex interactions.

More recently a number of studies have looked at hybrid Eulerian/Lagrangian solvers to compute vortical flows. These studies couple a RANS CFD solver to a particle-based Lagrangian method and employ a domain decomposition approach so that the CFD is only applied in the near-body region and the particle method is used in the outer region. Most methods also employ an 'overlap' region to couple the two methods where both solvers are used and the solutions are interpolated between each other. The first example of this is by Sitaraman [17] who coupled a RANS solver to a Particle Vorticity Transport Method (PVTM) to simulate rotor wakes. Anusonti-Inthra and Floros [18] extend this to a viscous PVTM to model the flow in isolated wing wakes. Zhao *et al.* [19] employ a viscous vortex particle method together with two different RANS solvers to investigate rotor wake flow. Stone *et al.* [20, 21] used an overset Unsteady RANS flow solver coupled to a vortex particle method to investigate rotor blade-vortex interactions. Pahla *et al.* [22] take a slightly different approach, whilst the domain is still decomposed into regions the Lagrangian method is applied to the entire domain whilst the Eulerian solver is only applied to the region close to the solid boundary. Essentially, they use the CFD to correct the particle method in the near-field.

In this work, we propose a new, hybrid method that couples a RANS solver with a Vortex Particle Method. The domain is not decomposed into regions; instead the VPM is used to approximate the flow solution and the CFD is used to calculate the difference between the approximation and the flowfield. In convection dominated flows, having a good approximation means that the diffusion normally found in CFD is eliminated. This is done by including the influence of the particles in the CFD solution through the Split Velocity Method, which has previously been used in the simulation of gusts [23, 24].

II. Methodology

The Navier-Stokes equations govern the motion of fluid flow. The equations are most commonly described using the Eulerian formulation. This is the approach used here within the CFD solver DLR-TAU [25], where the flow quantities are considered functions of the spatial location as they change with time. For vortex transit this requires a fine mesh throughout the computational domain leading to very high computational cost. However, the Split Velocity Method recently developed and implemented for gusts can be extended to produce an efficient method for vortex transit if a suitable alternative Lagrangian model of the vortex is available. The Vortex Particle Method solves the Navier-Stokes equations in velocity-vorticity form by considering the problem as a collection of fluid elements. The Split Velocity Method allows the vortex transit to be modelled by the Lagrangian scheme and the CFD can then compute the flow field

minus the model prediction. The key advantage is that coarser meshes can be used without the vortex being dissipated. This section first describes the CFD formulation and then presents the VPM. Finally, the coupling method is explained along with techniques used to speed up the computation.

A. CFD solver

In two dimensions, the unsteady Navier-Stokes equations are given by

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho E\end{bmatrix}+\frac{\partial}{\partial x}\begin{bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ u(\rho E+p)\end{bmatrix}+\frac{\partial}{\partial y}\begin{bmatrix}\rho v\\ \rho vu\\ \rho v^2+p\\ v(\rho E+p)\end{bmatrix}+\frac{\partial}{\partial x}\begin{bmatrix}0\\ \sigma_{xx}\\ \sigma_{xy}\\ u\sigma_{xx}+v\sigma_{xy}+q_x\end{bmatrix}+\frac{\partial}{\partial y}\begin{bmatrix}0\\ \sigma_{xy}\\ \sigma_{yy}\\ u\sigma_{xy}+v\sigma_{yy}+q_y\end{bmatrix}=0 \tag{1}$$

where the energy and pressure are given by

$$E=\frac{p}{\rho(\gamma-1)}+\frac{1}{2}(u^2+v^2) \tag{2}$$

and

$$p=(\gamma-1)(\rho E-\frac{\rho}{2}(u^2+v^2)) \tag{3}$$

respectively.

In this work the Navier-Stokes equations, Eq. (1), are solved using a cell-centred finite volume scheme on an unstructured mesh. The convective flux of the mean flow equations is approximated using a central difference scheme with matrix dissipation and the equations are integrated in time using a dual-timestepping method, where the time-derivative is discretised using a second-order Backward Difference Formula (BDF). Each time step is converged using the Backward Euler method in which the linear system is solved using the Lower-Upper Symmetric Gauss-Seidel (LUSGS) scheme. Although DLR-TAU employs a multigrid acceleration technique, it has been switched off for this work.

B. Vortex Particle Method

In a Vortex Particle Method, the flow is discretised into particles with concentrated circulations or strengths, which are then convected according to the Lagrangian form of the transport equation. Rather than using point vortices which have a singularity, vortex blobs or regularized vortex particles are used, which are particles with a finite core size. In this case, the vorticity field is given by

$$\omega(\mathbf{x},t) = \sum_{i=1}^{N} \Gamma_i(t)\zeta_{\sigma_i}(\mathbf{x} - \mathbf{x}_i(t)) \tag{4}$$

where ω is the vorticity, Γ_i is the strength of particle i, ζ_σ is the regularized smoothing kernel [26] and σ_i is the radius of particle i. In this work, all particles have the same radius so $\sigma_i = \sigma$.

For a two-dimensional case, the regularized smoothing kernel is given by

$$\zeta_\sigma = \frac{1}{\sigma^2}\zeta\left(\frac{|\mathbf{x}|}{\sigma}\right) \tag{5}$$

where $\zeta(\rho)$ is the smoothing or cutoff function. The cutoff function should be smooth and accurate [11] and several choices are listed in the paper by Wincklemans and Leonard [27]. In this work the second-order 2D gaussian smoothing, given by Eq. (6), is used.

$$\zeta(\rho) = \frac{1}{2\pi}\exp\left(-\frac{\rho^2}{2}\right) \tag{6}$$

The equations of motion for a regularized vortex particle method are given by

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{u}_\sigma(\mathbf{x}(t), t) \tag{7}$$

$$\frac{d}{dt}\Gamma(t) = \nu\nabla^2\omega V \tag{8}$$

The right hand side of Eq. (8) corresponds to a viscous diffusion where V is the volume asscoiated with a single particle. The velocity, $\mathbf{u}_\sigma(\mathbf{x}, t)$, is computed using the generalized Biot-Savart equation, which is

$$\mathbf{u}_\sigma(\mathbf{x}, t) = -\sum_{i=1}^{N} \mathbf{K}_\sigma(\mathbf{x} - \mathbf{x}_i(t))\,\Gamma_i(t). \tag{9}$$

From this point, the subscript σ is dropped from the velocity notation. The Biot-Savart equation, Eq. (9), is solved twice once for the VPM particle-induced velocities which will be denoted $\hat{\mathbf{u}}_p = [\hat{u}_p, \hat{v}_p]$, and correspond to the velocities that the particles induce on each other, and once for the CFD particle-induced velocities, denoted by $\hat{\mathbf{u}} = [\hat{u}, \hat{v}]$, which are the velocities the particles induce at the CFD grid nodes and are used for the Split Velocity Method. The convection equation, Eq. (7), is solved using a fourth order Runge-Kutta method and the velocity equation, Eq. (9), is solved using a Fast Multipole Method (FMM), which is described in section 3.

C. Coupling method

The CFD solver and the VPM are coupled together using the process shown in Fig. 1, which shows that the whole solution process involves many steps. The main steps in the coupling involve seeding the particles from the CFD vorticity and computing the CFD particle-induced velocities and source terms for the Split Velocity Method. This is described first, followed by a fast summation technique, which is used to reduce the computational cost of the process. This is necessary as the computation of the velocities and source terms requires summation over a large number of elements. Finally, the particle seeding routine is explained.

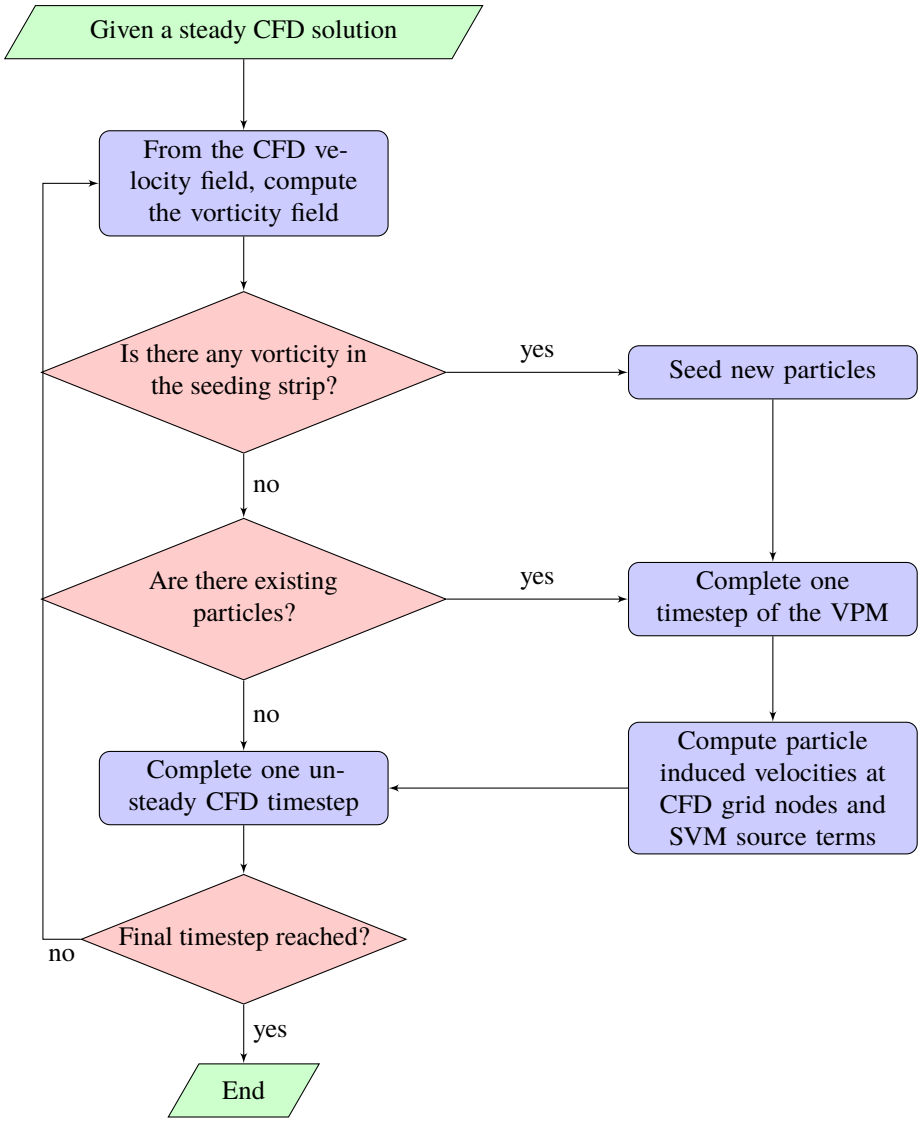


Fig. 1 Flowchart of the coupled CFD-VPM solver

1. Split Velocity Method

The formulation for the Split Velocity Method begins with the unsteady Navier-Stokes equations on a fixed mesh, given in Eq. (1). Then the velocity and energy are decomposed as

$$u = \tilde{u} + \hat{\hat{u}} \qquad v = \tilde{v} + \hat{\hat{v}} \qquad E = \tilde{E} + \hat{\tilde{E}} + \hat{\hat{E}} \tag{10}$$

where $\hat{\hat{u}}$ and $\hat{\hat{v}}$ are the induced vortex particle velocity components as explained above and $\hat{\hat{E}}$ is computed by substituting the velocity decompositions into Eq. (2), which after some manipulation gives

$$E = \underbrace{\frac{p}{\rho(\gamma-1)} + \frac{1}{2}(\tilde{u}^2 + \tilde{v}^2)}_{\tilde{E}} + \underbrace{(\tilde{u}\hat{\hat{u}} + \tilde{v}\hat{\hat{v}})}_{\hat{\tilde{E}}} + \underbrace{\frac{1}{2}(\hat{\hat{u}}^2 + \hat{\hat{v}}^2)}_{\hat{\hat{E}}} \tag{11}$$

The pressure remains unchanged and is therefore given by Eq. (3). The Navier-Stokes equations for the Split Velocity Method are then obtained by substituting the decompositions in Eq. (10) into the unsteady Navier-stokes equations given by Eq. (1). This gives

$$\begin{aligned}
&\frac{\partial}{\partial t}\begin{bmatrix} \rho \\ \rho(\tilde{u}+\hat{\hat{u}}) \\ \rho(\tilde{v}+\hat{\hat{v}}) \\ \rho(\tilde{E}+\hat{\tilde{E}}+\hat{\hat{E}}) \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} \rho(\tilde{u}+\hat{\hat{u}}) \\ \rho(\tilde{u}+\hat{\hat{u}})^2 + p \\ \rho(\tilde{u}+\hat{\hat{u}})(\tilde{v}+\hat{\hat{v}}) \\ (\rho(\tilde{E}+\hat{\tilde{E}}+\hat{\hat{E}}))(\tilde{u}+\hat{\hat{u}}) \end{bmatrix} \\
&+ \frac{\partial}{\partial y}\begin{bmatrix} \rho(\tilde{v}+\hat{\hat{v}}) \\ \rho(\tilde{v}+\hat{\hat{v}})(\tilde{u}+\hat{\hat{u}}) \\ \rho(\tilde{v}+\hat{\hat{v}})^2 + p \\ (\rho(\tilde{E}+\hat{\tilde{E}}+\hat{\hat{E}}))(\tilde{v}+\hat{\hat{v}}) \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} 0 \\ \sigma_{xx} \\ \sigma_{xy} \\ (\tilde{u}+\hat{\hat{u}})\sigma_{xx} + (\tilde{v}+\hat{\hat{v}})\sigma_{xy} + q_x \end{bmatrix} + \frac{\partial}{\partial y}\begin{bmatrix} 0 \\ \sigma_{xy} \\ \sigma_{yy} \\ (\tilde{u}+\hat{\hat{u}})\sigma_{xy} + (\tilde{v}+\hat{\hat{v}})\sigma_{yy} + q_y \end{bmatrix} \\
&= 0
\end{aligned} \tag{12}$$

where,

$$\sigma_{xx} = \tfrac{2}{3}\tfrac{\mu}{Re}\left(2\tfrac{\partial \tilde{u}}{\partial x} - \tfrac{\partial \tilde{v}}{\partial y}\right) + \tfrac{2}{3}\tfrac{\mu}{Re}\left(2\tfrac{\partial \hat{\hat{u}}}{\partial x} - \tfrac{\partial \hat{\hat{v}}}{\partial y}\right) \quad \sigma_{yy} = \tfrac{2}{3}\tfrac{\mu}{Re}\left(2\tfrac{\partial \tilde{v}}{\partial y} - \tfrac{\partial \tilde{u}}{\partial x}\right) + \tfrac{2}{3}\tfrac{\mu}{Re}\left(2\tfrac{\partial \hat{\hat{v}}}{\partial y} - \tfrac{\partial \hat{\hat{u}}}{\partial x}\right)$$

$$\sigma_{xy} = \frac{\mu}{Re}\left(\frac{\partial \tilde{u}}{\partial y} + \frac{\partial \tilde{v}}{\partial x}\right) + \frac{\mu}{Re}\left(\frac{\partial \hat{\hat{u}}}{\partial y} + \frac{\partial \hat{\hat{v}}}{\partial x}\right)$$

$$q_x = -\tfrac{\mu}{Pr}\tfrac{1}{Re}\tfrac{1}{(\gamma-1)M_\infty^2}\tfrac{\partial T}{\partial x} \quad q_y = -\tfrac{\mu}{Pr}\tfrac{1}{Re}\tfrac{1}{(\gamma-1)M_\infty^2}\tfrac{\partial T}{\partial y}$$

$$T = \frac{\gamma M_\infty^2 p}{\rho}$$

where, Pr, μ and Re are the Prandtl number, dynamic viscosity and Reynolds number respectively. Separating the induced velocities from the rest of the solution and after some manipulation of the terms, the Navier-Stokes equations are given as

$$\frac{\partial}{\partial t}\begin{bmatrix} \rho \\ \rho\tilde{u} \\ \rho\tilde{v} \\ \rho\tilde{E} \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} \rho(\tilde{u}+\hat{\hat{u}}) \\ \rho\tilde{u}(\tilde{u}+\hat{\hat{u}})+p \\ \rho\tilde{v}(\tilde{u}+\hat{\hat{u}}) \\ \rho\tilde{E}(\tilde{u}+\hat{\hat{u}})+p\tilde{u} \end{bmatrix} + \frac{\partial}{\partial y}\begin{bmatrix} \rho(\tilde{v}+\hat{\hat{v}}) \\ \rho\tilde{u}(\tilde{v}+\hat{\hat{v}}) \\ \rho\tilde{v}(\tilde{v}+\hat{\hat{v}})+p \\ \rho\tilde{E}(\tilde{v}+\hat{\hat{v}})+p\tilde{v} \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} 0 \\ \sigma_{xx} \\ \sigma_{xy} \\ \tilde{u}\sigma_{xx}+\tilde{v}\sigma_{xy}+q_x \end{bmatrix} + \frac{\partial}{\partial y}\begin{bmatrix} 0 \\ \sigma_{xy} \\ \sigma_{yy} \\ \tilde{u}\sigma_{xy}+\tilde{v}\sigma_{yy}+q_y \end{bmatrix} + \begin{bmatrix} 0 \\ s_m(\hat{\hat{u}}) \\ s_m(\hat{\hat{v}}) \\ s_e(\hat{\hat{u}},\hat{\hat{v}}) \end{bmatrix} = 0. \tag{13}$$

The source terms are given by

$$s_m(\cdot) = \rho\left\{\frac{\partial \cdot}{\partial t} + (\tilde{u}+\hat{\hat{u}})\frac{\partial \cdot}{\partial x} + (\tilde{v}+\hat{\hat{v}})\frac{\partial \cdot}{\partial y}\right\} \tag{14}$$

$$\begin{aligned} s_e(\hat{\hat{u}},\hat{\hat{v}}) &= \tilde{u}s_m(\hat{\hat{u}}) + \tilde{v}s_m(\hat{\hat{v}}) + p\left[\frac{\partial \hat{\hat{u}}}{\partial x} + \frac{\partial \hat{\hat{v}}}{\partial y}\right] \\ &+ \sigma_{xx}\frac{\partial \hat{\hat{u}}}{\partial x} + \sigma_{yy}\frac{\partial \hat{\hat{v}}}{\partial y} + \sigma_{xy}\left[\frac{\partial \hat{\hat{v}}}{\partial x} + \frac{\partial \hat{\hat{u}}}{\partial y}\right] \end{aligned} \tag{15}$$

It is noted that the stress tensors, σ, in SVM are calculated based on velocity derivatives for total velocities u and v meaning that they include the induced vortex particle velocities for the calculation of viscous fluxes. This is to eliminate the introduction of dissipative source terms arising from separating velocity derivatives.

The requirement for the particle induced velocities to be included in the CFD as grid velocities means that the velocity equation, Eq. (9) must be evaluated at all M nodes of the CFD grid for all N particles. The computational cost of evaluating this equation directly is $O(NM)$. Furthermore, computing the source terms, Eqs. (14) and (15), is a further $O(NM)$ operation if evaluated directly. This is computationally the most expensive part of the hybrid solver as the source terms must be computed at every iteration of the dual-timestepping CFD solver. The Fast Multipole Method (FMM) [28] allows the cost of the velocity computation to be reduced to $O(N)$.

2. Conservation of the scheme

The use of the Split Velocity Method along with a Vortex Particle Method to preserve vortices is a conservative method as long as the source terms that arise in the SVM are integrated in the same way as the rest of the RANS equations. As such, as long as this is done, then it is still possible to write the full equations in integral form. The role of the Vortex Particle Method is to provide the decomposition for the velocities used in the Split Velocity Method. Therefore, the velocities calculated using the Vortex Particle Method have no effect on the conservative nature of the scheme, although their accuracy is cleary important for preserving the vorticity. However, it should be noted that whilst this is true for the results of the simulations as they are run on the coarse mesh. The figures presented in section III are the coarse mesh results after they have been interpolated onto the fine mesh, which has been carried out for clearer visualisation of the vortices.

3. Fast Multipole Method

The FMM has been applied to the computation of the particle-induced velocity computation, Eq. (9) and the source terms, Eqs. (14) and (15). First the formulation for the velocities is presented where a distinction between sources and targets is made. Sources are always the particles whilst the targets are the particles if the VPM particle-induced velocities, $\hat{\mathbf{u}}_p$, are being calculated or the CFD grid nodes if the CFD particle-induced velocities, $\hat{\mathbf{u}}$, are being calculated.

For a particle simulation, following the formulation presented by Cruz and Barba [29], the complex conjugate velocity is given by the multipole expansion

$$u(y)^* \approx \sum_{m=0}^{p-1} \left(\sum_{i=1}^{N} c_i a_m(x_i, x_c) \right) f_m(y, x_c) \tag{16}$$

where the superscript $*$ corresponds to the complex conjugate, y is a target point, x_i is a source point, x_c is the centre of a cluster of source points and p is a truncation parameter. In Eq. (16), c_i is given by

$$c_i = -\frac{j\Gamma_i}{2\pi} \tag{17}$$

where j is the unit imaginary number, a_m is given by

$$a_m(x_i, x_c) = (x_i - x_c)^m \tag{18}$$

and f_m is given by

$$f_m(y, x_c) = (y - x_c)^{-m-1} \tag{19}$$

The version of the FMM used in this work begins with a decomposition of the spatial domain into an adaptive quadtree

and follows the algorithm for the two-dimensional adaptive FMM presented in Carrier *et al.* [30].

The source terms given by Eq. (14) can be calculated by the chain rule so that, for example, $s_m(\hat{\tilde{u}})$ is computed from

$$\begin{aligned}\rho\left\{\frac{\partial\hat{\tilde{u}}}{\partial t}+(\tilde{u}+\hat{\tilde{u}})\frac{\partial\hat{\tilde{u}}}{\partial x}+(\tilde{v}+\hat{\tilde{v}})\frac{\partial\hat{\tilde{u}}}{\partial y}\right\} &= \rho\left\{\frac{\partial\hat{\tilde{u}}}{\partial x}\frac{\partial x}{\partial t}+\frac{\partial\hat{\tilde{u}}}{\partial y}\frac{\partial y}{\partial t}+\frac{\partial\hat{\tilde{u}}}{\partial\Gamma}\frac{\partial\Gamma}{\partial t}\right.\\ &\quad\left.+(\tilde{u}+\hat{\tilde{u}})\frac{\partial\hat{\tilde{u}}}{\partial x}+(\tilde{v}+\hat{\tilde{v}})\frac{\partial\hat{\tilde{u}}}{\partial y}\right\}\end{aligned} \tag{20}$$

Noting that

$$\frac{\partial x}{\partial t}=\hat{\tilde{u}}_p \qquad \frac{\partial y}{\partial t}=\hat{\tilde{v}}_p \tag{21}$$

gives the source term as

$$\begin{aligned}\rho\left\{\frac{\partial\hat{\tilde{u}}}{\partial t}+(\tilde{u}+\hat{\tilde{u}})\frac{\partial\hat{\tilde{u}}}{\partial x}+(\tilde{v}+\hat{\tilde{v}})\frac{\partial\hat{\tilde{u}}}{\partial y}\right\} &= \rho\left\{\frac{\partial\hat{\tilde{u}}}{\partial x}\hat{\tilde{u}}_p+\frac{\partial\hat{\tilde{u}}}{\partial y}\hat{\tilde{v}}_p+\frac{\partial\hat{\tilde{u}}}{\partial\Gamma}\frac{\partial\Gamma}{\partial t}\right.\\ &\quad\left.+(\tilde{u}+\hat{\tilde{u}})\frac{\partial\hat{\tilde{u}}}{\partial x}+(\tilde{v}+\hat{\tilde{v}})\frac{\partial\hat{\tilde{u}}}{\partial y}\right\}\end{aligned} \tag{22}$$

The computation of $\frac{\partial\hat{\tilde{u}}}{\partial\Gamma}\frac{\partial\Gamma}{\partial t}$ is simplified by utilising the fact that the CFD uses a second-order backward difference dual-timestepping scheme. This means that $\frac{\partial\Gamma}{\partial t}$ can be formulated as

$$\frac{\partial\Gamma}{\partial t}=\frac{3\Gamma^{n+1}}{2\Delta t}-\frac{4\Gamma^{n}}{2\Delta t}+\frac{\Gamma^{n-1}}{2\Delta t} \tag{23}$$

This results in

$$\frac{\partial\hat{\tilde{u}}}{\partial\Gamma}\frac{\partial\Gamma}{\partial t}=\hat{\tilde{u}}^{n+1}\frac{3}{2\Delta t}-\hat{\tilde{u}}^{n}\frac{4}{2\Delta t}+\hat{\tilde{u}}^{n-1}\frac{1}{2\Delta t} \tag{24}$$

and $\hat{\tilde{u}}^{n+1}$, $\hat{\tilde{u}}^{n}$ and $\hat{\tilde{u}}^{n-1}$ can be calculated using the FMM and Eq. (16) by substituting Γ^{n+1}, Γ^{n} and Γ^{n-1} for Γ, respectively. Finally the $\frac{\partial\hat{\tilde{\mathbf{u}}}}{\partial\mathbf{x}}$ terms are calculated using

$$\frac{\partial u(y)^*}{\partial\mathbf{x}}\approx\sum_{m=0}^{p-1}\left(\sum_{i=1}^{N}c_i a_m(x_i,x_c)\right)\nabla f_m(y,x_c) \tag{25}$$

So the source term given by Eq. (20) is calculated by computing Eq. (24) and Eq. (25) and then substituting these into Eq. (20). The source terms involving the other induced velocity components are calculated in a similar way.

4. Particle seeding

Vortex particles are initialised or "seeded" according to the vorticity in the CFD solution. A region in space, a line in 2D, is identified and particles are seeded where any vorticity is present in this region. Using the 2D case, shown in Fig. 2, as the example, the user specifies the length of the line and the position along the x-axis. The particle spacing, h

is set to

$$h = u_{ref}\Delta t \tag{26}$$

so that no vorticity from the CFD is missed. The number of particles in the y-direction, N_y, is set by

$$N_y = \frac{y_{max} - y_{min}}{h} \tag{27}$$

so that each particle lies in squares of equal area, h^2. The particle radius, σ, is then calculated by

$$\sigma = \frac{h}{\beta} \tag{28}$$

where β is the particle overlap ratio, which should be less than 1 to ensure that particles overlap [31]. The vorticity at the centre of each particle is interpolated from the vorticity at the CFD nodes using an inverse distance weighting so that for particle i the vorticity is given by

$$\omega_i(\mathbf{x}) = \begin{cases} \frac{\sum_{j=1}^{N} s_j(\mathbf{x})\omega_j}{\sum_{j=1}^{N} s_j(\mathbf{x})} & \text{if } d(\mathbf{x}, \mathbf{x}_j) \neq 0 \text{ for all } j \\ \omega_j & \text{if } d(\mathbf{x}, \mathbf{x}_j) = 0 \text{ for some } j \end{cases} \tag{29}$$

In this equation

$$s_j(\mathbf{x}) = \frac{1}{d(\mathbf{x}, \mathbf{x}_j)^p} \tag{30}$$

where d is the distance metric and p is the power parameter. Using a larger value of p has the effect of increasing the influence of solution from the closest CFD points. In this work, a value of $p = 7$ was used as smaller values resulted in highly unsmooth vorticity distributions, which eventually results in the method breaking. Now each particle has an associated vorticity the strength of the particles that gives this vorticity distribution must be found. In almost all the literature the particle strengths are initialised by multiplying the particle vorticity by the volume of the particle, i.e.

$$\Gamma_i = \omega_i h^d, \tag{31}$$

where d is the dimension of the problem. This introduces an error, known as a regularisation or smoothing error [32], which for most Vortex Particle Methods that appear in the literature, appears to be acceptable. However, in this work, this means that the core vorticity is not captured entirely and this has the effect of dissipating the vortex. An alternative is to solve Eq. (4), which can be written as

$$\mathbf{A\Gamma} = \omega, \tag{32}$$

where

$$A_{ij} = \zeta_\sigma(\mathbf{x}_j - \mathbf{x}_i). \tag{33}$$

The condition number of the matrix **A** is dependent on the value of σ, which itself is dependent on the inter-particle spacing and the overlap ratio. The condition number of the matrix correlates with σ, so in order to achieve a better conditioned system with the inter-particle spacing can be reduced by increasing the number of particles or the overlap ratio can be increased. Unfortunately, the VPM requires the overlap ratio to be less than one to ensure convergence [31] and computational resources restrict the number of particles that can be seeded on each timestep, especially when simulating long times. Therefore, a way of solving this ill-conditioned system is needed. This system is solvable using Singular Value Decomposition (SVD) but results in an odd-even coupling of the particle strengths. This then causes numerical errors when the velocities are computed. Therefore, the problem becomes one of solving Eq. (32) for smooth particle strengths. This was solved by adding fourth order dissipation to the equation to damp out the high frequencies.

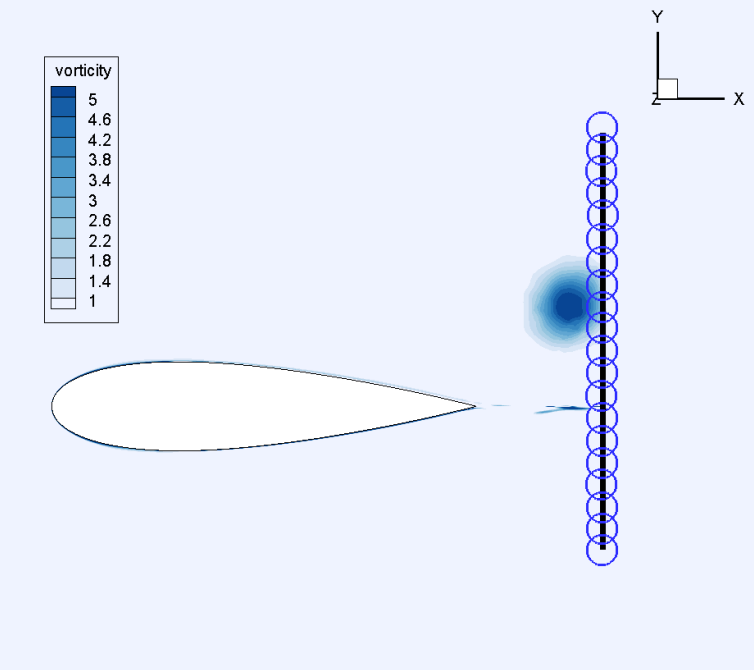


Fig. 2 Example of particles being seeded in 2D.

III. Results

In this section the coupled solver is demonstrated. For all the results presented, the coupled CFD-VPM approach was only applied to the coarse mesh but all simulations were also performed on both meshes using the CFD solver on its own. The results for the coupled simulation were produced by summing the vorticity produced by the CFD with the vorticity produced by the vortex particles and then interpolating this onto the fine mesh.

A. Blade-vortex interaction

The first test case involved the interaction of a NACA 0012 airfoil with a vortex given analytically by the Scully vortex model [33]. This case has been simulated by many authors [34–36] and here the results from the CFD solver on both the fine mesh and the coarse mesh as well as the coupled solver on the coarse mesh are compared, along with the results presented in the literature. The case is run with a freestream Mach number of 0.8 and Reynolds number of 3.6×10^6. The vortex was defined with a core radius of 0.05 and non-dimensional strength of -0.2, where the non-dimensionalization is performed with respect to the chord length. The vortex is initially located 5 chord lengths upstream of the airfoil leading edge and 0.26 chord lengths beneath the airfoil chordline. All the simulations were performed with a physical timestep size of 0.001 seconds. The fine mesh is shown in Fig. 3; it was designed with heavy refinement along the path of the vortex, which resulted in a mesh with 173016 points. The coarse mesh is shown in Fig. 4 and contains 26758 points. For this case, particles were not seeded as described in section 4, instead a complete set of 400 particles were initialised at the beginning of the simulation according to the analytical description of the Scully vortex and these were allowed to convect downstream using the VPM whilst their influence was included in the CFD through the SVM. This number of particles was chosen as it corresponds to the minimum square number that allows the strength of the vortex to be captured. Employing fewer particles results in a smaller response in the lift coefficient as the vortex is weaker whilst more particles results in a marginal increase in accuracy but also increase the computational cost.

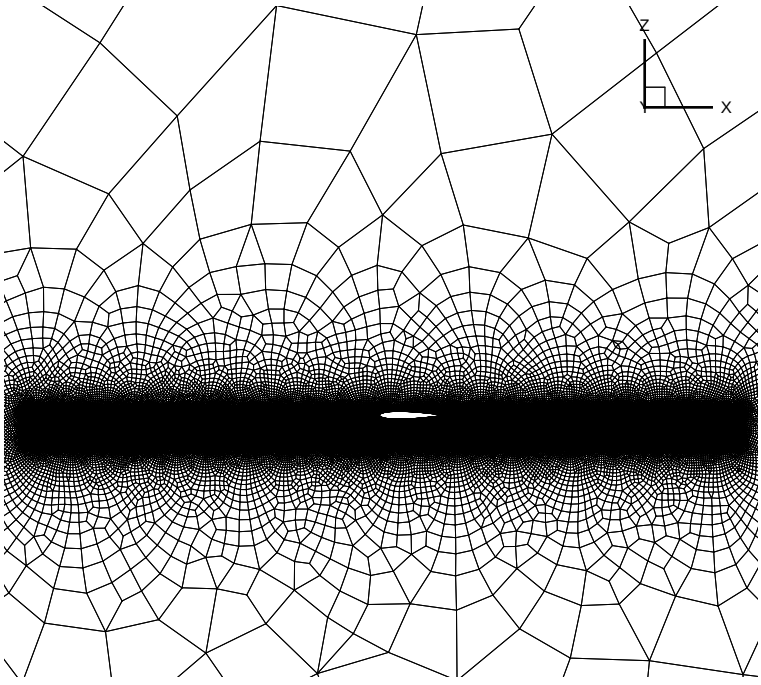


Fig. 3 Fine mesh for Blade Vortex Interaction

Figure 5 shows the result of lift coefficient as the vortex traverses the domain for all three methods. Clearly, the CFD

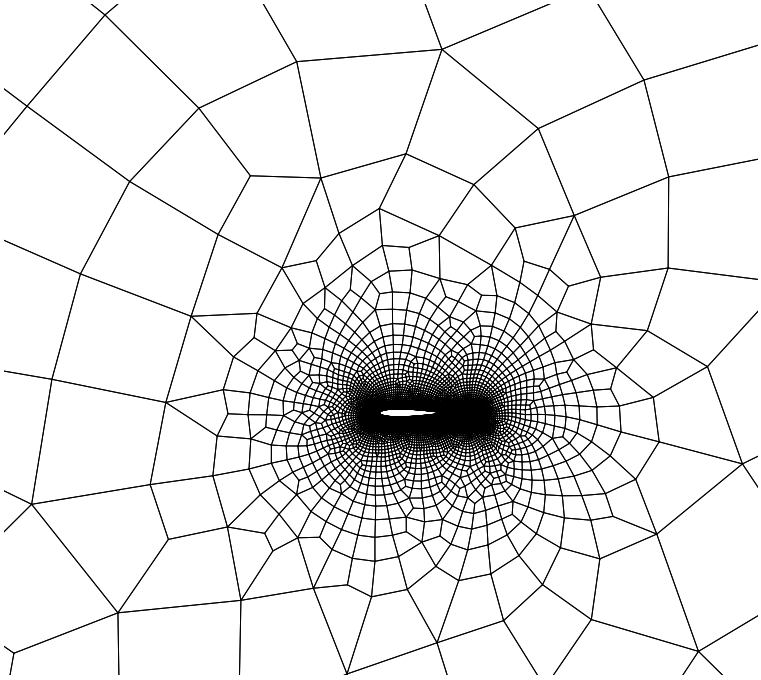


Fig. 4 Coarse mesh for Blade Vortex Interaction

solver on the coarse mesh severely dissipates the vortex whereas the CFD solver on the fine mesh and the coupled solver on the coarse mesh show good agreement with each other and also with results from the literature. There is a slight difference in the lift coefficient value after the vortex passes the airfoil. This could be due to the interference from the wake that occurs when using the CFD solver on the fine mesh, which is not present in the coupled solution as the airfoil wake is not captured on the coarse mesh. This can most clearly be seen in Fig. 6 which shows the vorticity contour for the CFD solver on the fine mesh and the coupled solver on the coarse mesh for the vortex at different positions. Figure 6 shows that the vorticity matches for both the CFD solver on the fine mesh and the coupled solver on the coarse mesh. However, due to the presence of the wake in the fine mesh this leads to quite large errors in vorticity between the fine and coupled solutions even when the vortex is just at the leading edge. At $X_v = 0.0$ the L2 error in vorticity between the fine and coupled solution is $156.36s^{-1}$ and the maximum error is $605.37s^{-1}$, but this value is located in the wake which is not captured by the coarse mesh and therefore not captured by the coupled solver. Discounting the wake values the L2 error is $30.58s^{-1}$ and the maximum error is $5.00s^{-1}$. The coupled solver simulation took 145 minutes to run whereas the fine simulation took 367 minutes giving a CPU speedup of 2.53.

B. NACA 0021

This test case involves the interaction of an artificially generated vortex with a NACA 0021 airfoil and is used to test the vortex particle seeding. The two meshes used for this simulation both had the same point distribution in the airfoil near field but one of the meshes, the coarse mesh shown in Fig. 7, coarsened away from this point whilst the other, the

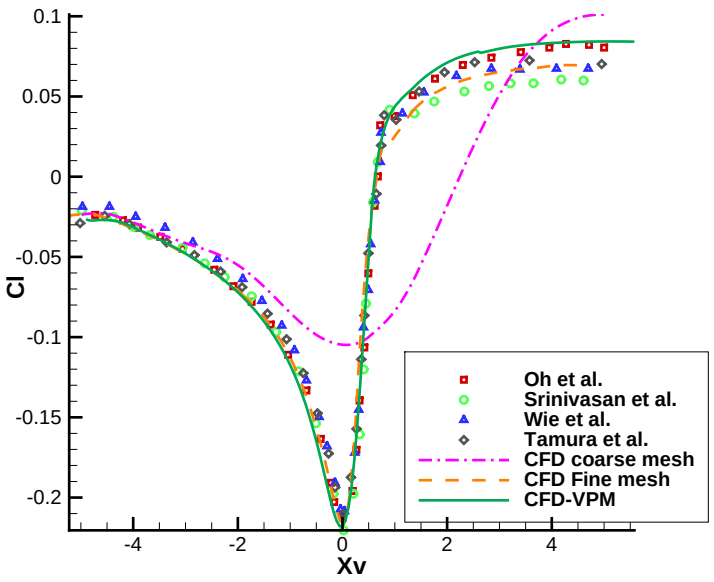


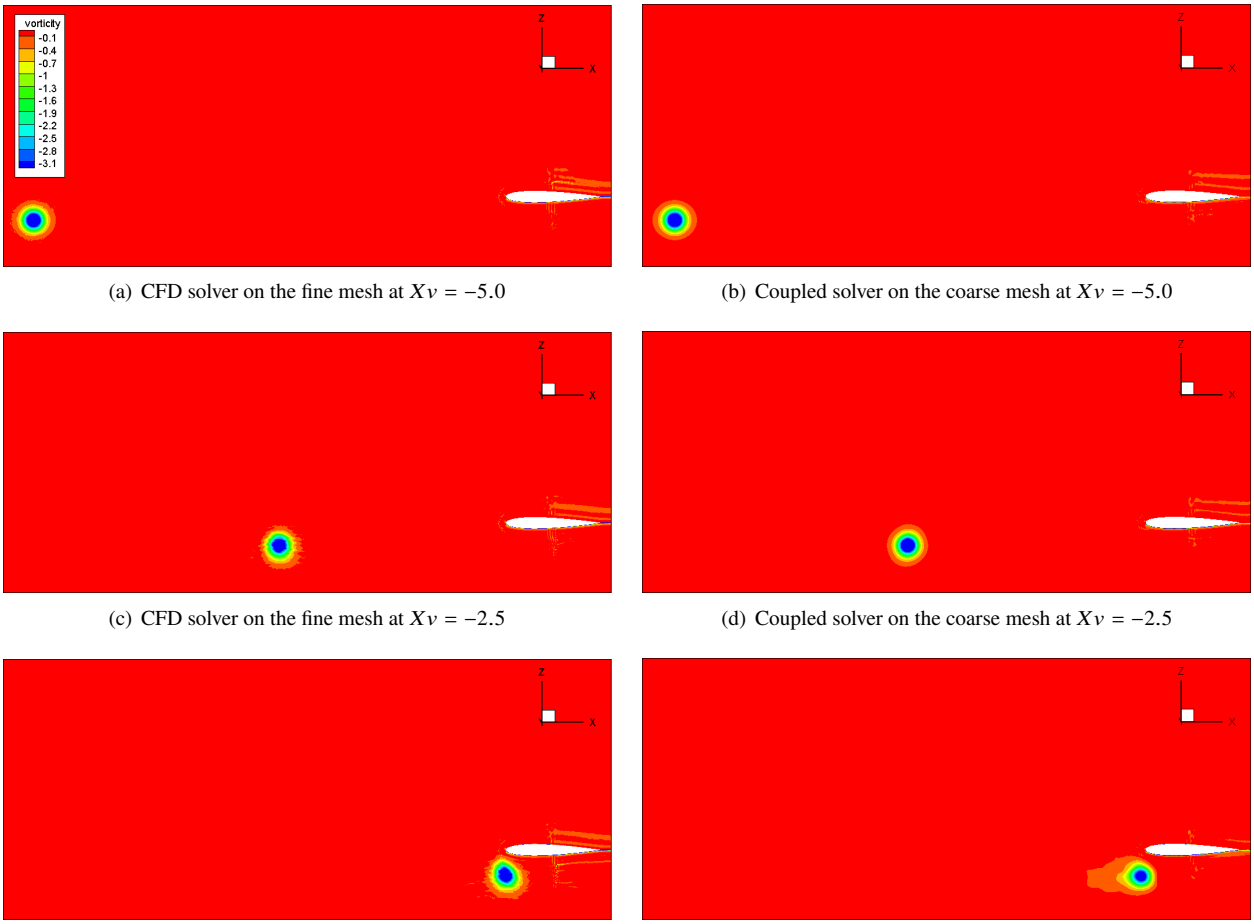
Fig. 5 Comparison of the lift variation during BVI

fine mesh shown in Fig. 8, maintained this node density along the vortex path. The fine mesh has a total of 139,810 points whilst the coarse mesh consists of 19,012 points.

The vortex was initially inserted into the CFD and allowed to convect through the mesh beyond the trailing edge. All the simulations were performed using a timestep size of 1.0×10^{-4} seconds and were run for 150 timesteps.

For the coupled solver, the particle seeding line was placed at 30% chord downstream of the trailing edge, as indicated by the dashed line in Fig. 7, and was sized vertically so it was slightly larger than the vortex. The coupled solver was set up to attempt to seed particles, with an overlap ratio of 0.5, at every time step. The result was that 22 particles were seeded every timestep until the 56th timestep by which point the vortex had passed through the seeding line completely. This meant that the computation reached a maximum of 1210 particles.

Figures 9, 10 and 11 show the vorticity distribution at four different timesteps for the CFD-only computation on the fine mesh, CFD-only computation on the coarse mesh and coupled computation on the coarse mesh respectively. As can be seen the coarse mesh is highly dissipative for the CFD-only computation but by coupling a VPM to the CFD solver the vortex is preserved. Furthermore, the CFD-only simulation on the fine mesh took 35% longer than the simulation using the coupled solver meaning the coupled solver is computationally more efficient even though there has not been effort to make the VPM solver highly efficient. The particles at each of these timesteps can be seen in Fig. 12. This shows how the number of particles changes at each time and how they wrap up to the shape of the vortex as the simulation progresses.



(a) CFD solver on the fine mesh at $Xv = -5.0$

(b) Coupled solver on the coarse mesh at $Xv = -5.0$

(c) CFD solver on the fine mesh at $Xv = -2.5$

(d) Coupled solver on the coarse mesh at $Xv = -2.5$

(e) CFD solver on the fine mesh at $Xv = 0.0$

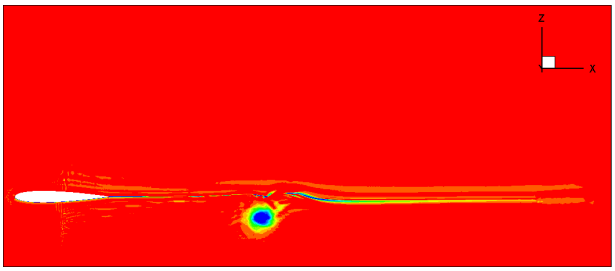
(f) Coupled solver on the coarse mesh at $Xv = 0.0$

Fig. 6 Comparison of vorticity contour during BVI

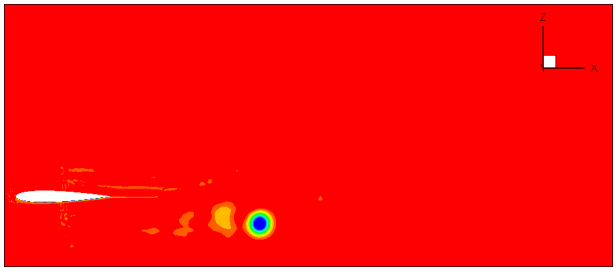
C. Tandem airfoil

This last test case involved two NACA 0021 airfoils in a tandem configuration. The fore airfoil is pitched down by 10 degrees which causes it to shed a vortex from the trailing edge. This vortex then convects downstream where it interacts with the aft airfoil. The two meshes used in this case are shown in Figs. 13 and 14 for the coarse mesh, with 163952 points, and fine mesh, with 256312 points. The pitching motion occurred over 100 timesteps and the simulations were run with a timestep size of 1×10^{-4} seconds. The freestream conditions corresponded to a Mach number of 0.15 and a Reynolds number of 2×10^{6}. The particles were seeded every timestep at a position 60% chord downstream of the trailing edge of the first airfoil. The size of the seeding line was dimensioned such that 117 particles were seeded on each timestep and the nature of the simulation, with the constant trailing wake from the fore airfoil meant that the simulation seeded particles for the whole duration of the simulation.

The comparison of vorticity contours between the CFD-only on the fine mesh and the coupled solver on the coarse mesh at different times of the simulation are shown in Fig. 15. The results of the CFD-only simulation on the coarse



(g) CFD solver on the fine mesh at $Xv = 2.5$



(h) Coupled solver on the coarse mesh at $Xv = 2.5$

(i) CFD solvr on the fine mesh at $Xv = 5.0$

(j) Coupled solver on the coarse mesh at $Xv = 5.0$

Fig. 6 (Cont.) Comparison of vorticity contour during BVI

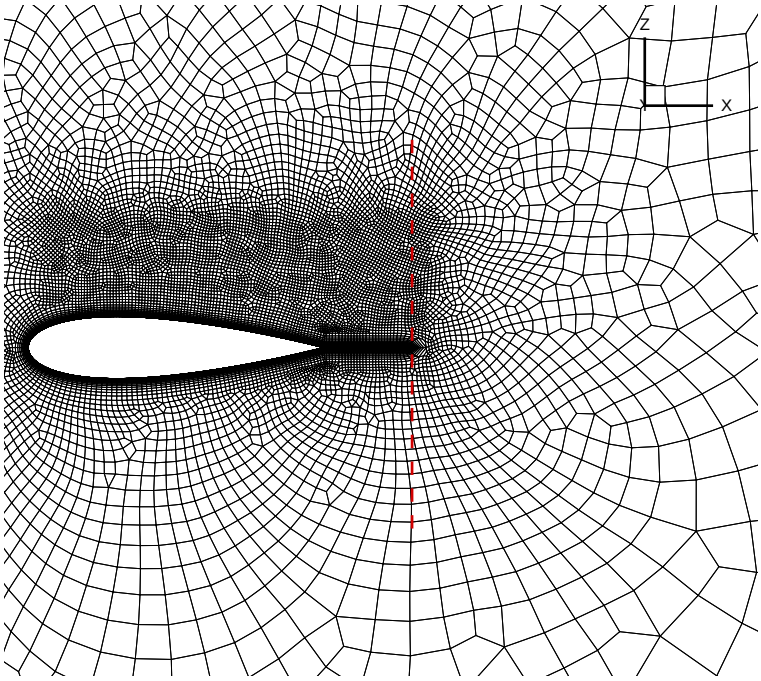


Fig. 7 Coarse mesh used in NACA 0021 test case consisting of 19012 points

mesh are not presented but the results echo the findings of the previous two cases, where the coarse mesh is highly dissipative of the vortex. However, in this case, as shown in Fig. 15, it appears that the coupled solver gives a better solution than even the CFD solver on the fine mesh as the vortex seems more clearly defined in the later stages of the simulation. Figure 16 shows the variation in lift coefficient over the trailing aerofoil as the simulation progresses for both

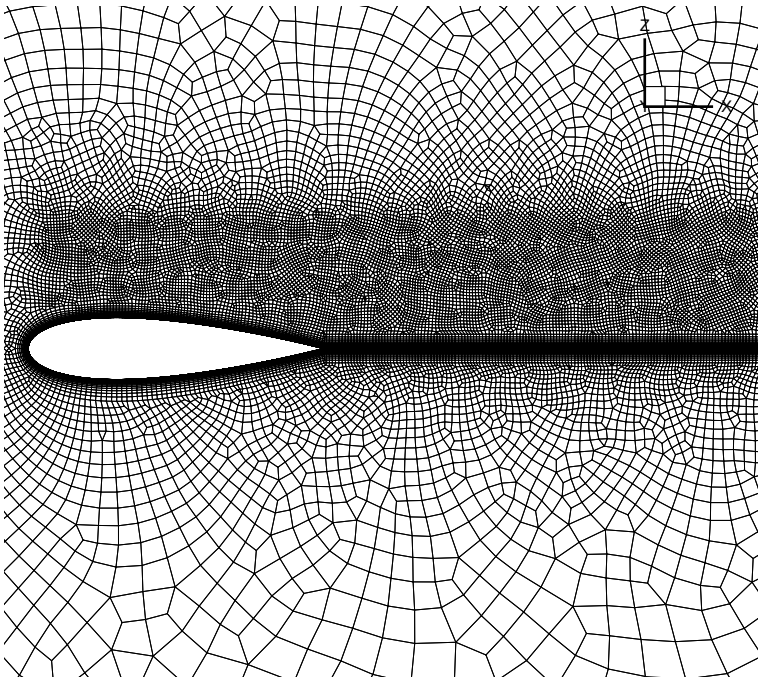
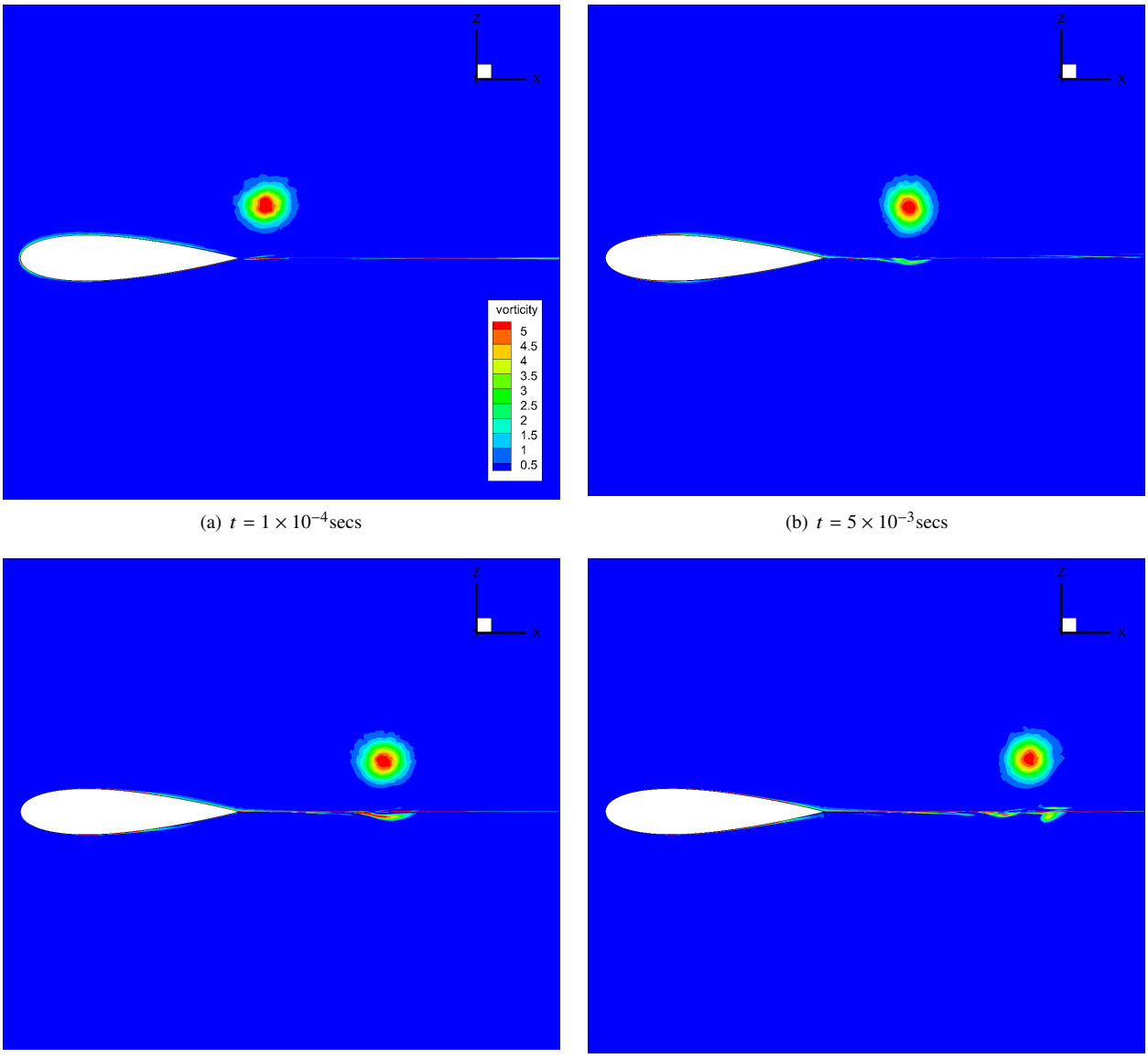


Fig. 8 Fine mesh used in NACA 0021 test case consisting of 139810 points

the coupled solver and the fine solution. It can be seen that as soon as particles are created there is a slight difference in the lift coefficient of the trailing aerofoil but the largest difference between the two solutions occurs as the vortex is passing over the top of the front portion of the aerofoil. The error here is maximum and has a value of 0.051. The speedup achieved for this simulation is 1.68. The reason for this much lower value, than was achieved in the first test case, is due to the fact that the simulation continues to initialise particles even once the vortex has been shed. This is something that could be rectified by employing conditional seeding and hence the speedup could be increased for this particlular case.

IV. Conclusion

A coupled CFD-VPM solver has been created. It has been shown to preserve the vortex structure on a coarse CFD mesh whilst also being computationally more efficient compared to running the same simulation using the CFD solver alone on a fine mesh. The computational cost of the method has been reduced by employing a Fast Multipole Method to compute the induced velocities and source terms necessary for the Split Velocity Method, which is used to couple the two solvers. The next steps in this work would be to introduce a particle merging scheme and parallelize the code to enable more realistic and meaningful simulations to be performed in an adequate timeframe.



(a) $t = 1 \times 10^{-4}$secs

(b) $t = 5 \times 10^{-3}$secs

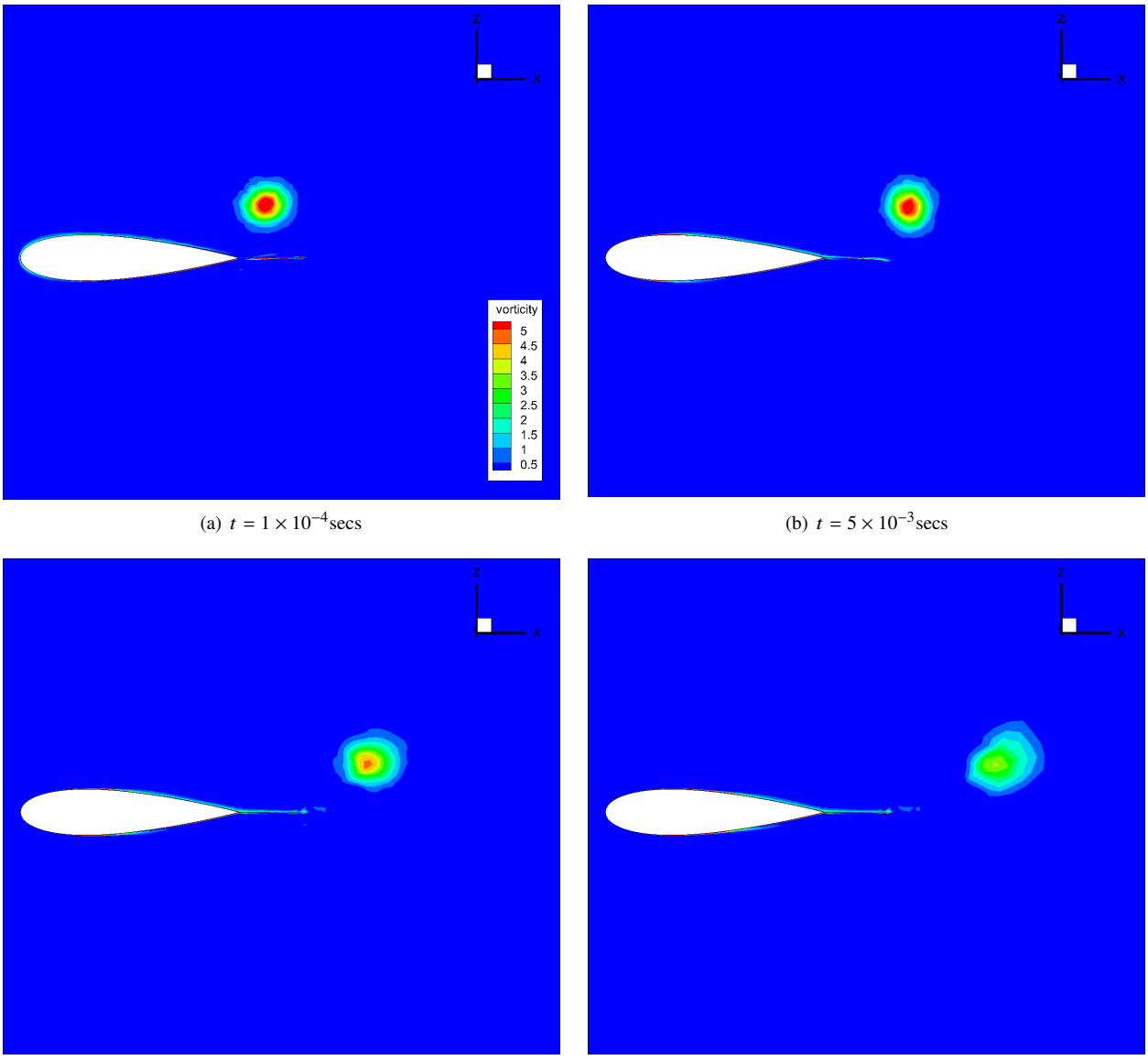
(c) $t = 1 \times 10^{-2}$secs

(d) $t = 1.5 \times 10^{-2}$secs

Fig. 9 Vorticity contour of NACA 0021 tip vortex simulation using the CFD solver only on the fine mesh

Funding Sources

The research leading to these results was co-funded by Innovate UK, the UK's innovation agency, within the Enhanced Fidelity Transonic Wing project.



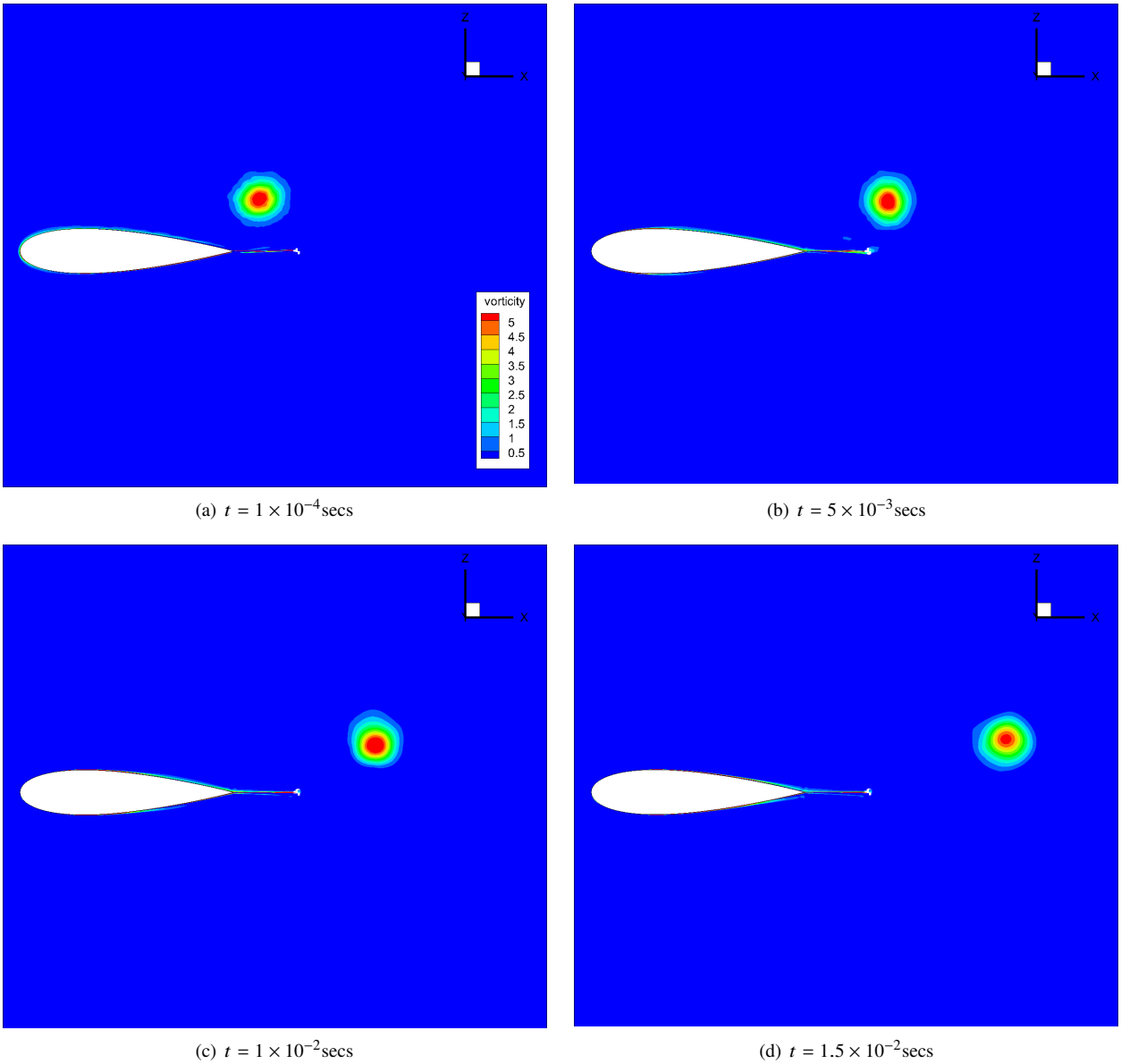
(a) $t = 1 \times 10^{-4}$secs

(b) $t = 5 \times 10^{-3}$secs

(c) $t = 1 \times 10^{-2}$secs

(d) $t = 1.5 \times 10^{-2}$secs

Fig. 10 Vorticity contour of NACA 0021 tip vortex simulation using the CFD solver only on the coarse mesh



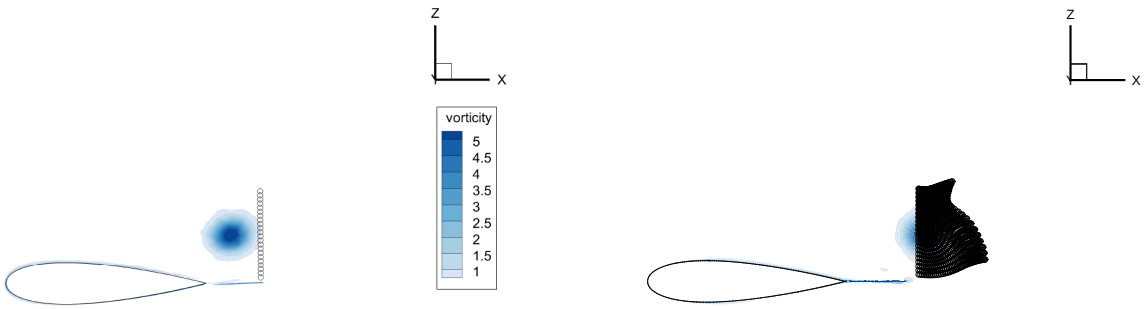
(a) $t = 1 \times 10^{-4}$secs

(b) $t = 5 \times 10^{-3}$secs

(c) $t = 1 \times 10^{-2}$secs

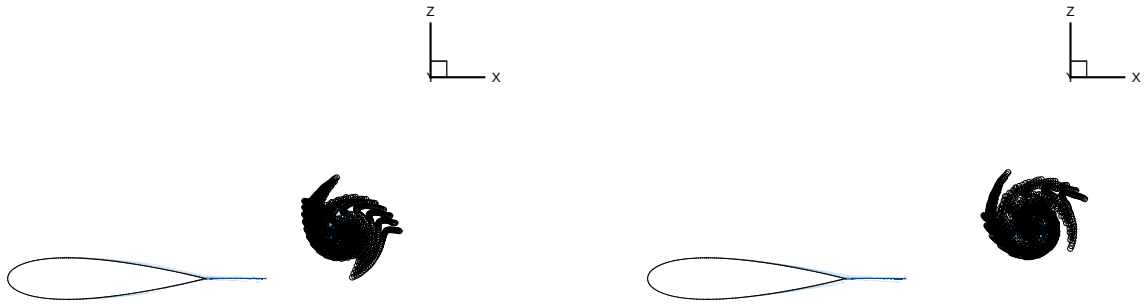
(d) $t = 1.5 \times 10^{-2}$secs

Fig. 11 Vorticity contour of NACA 0021 tip vortex simulation using the coupled CFD-VPM solver on the coarse mesh



(a) $t = 1 \times 10^{-4}$secs

(b) $t = 5 \times 10^{-3}$secs



(c) $t = 1 \times 10^{-2}$secs

(d) $t = 1.5 \times 10^{-2}$secs

Fig. 12 Particle visualisation as the simulation progresses

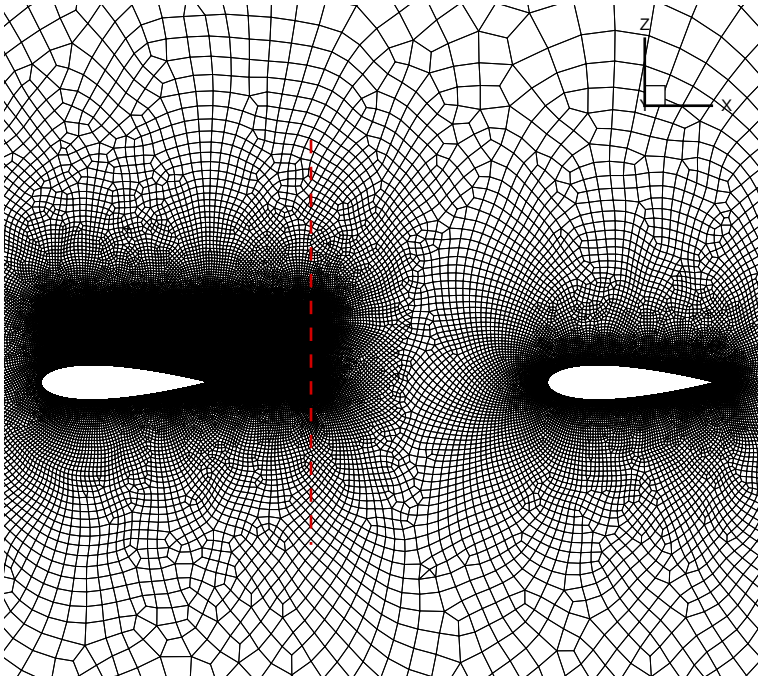


Fig. 13 Coarse mesh for tandem airfoil case

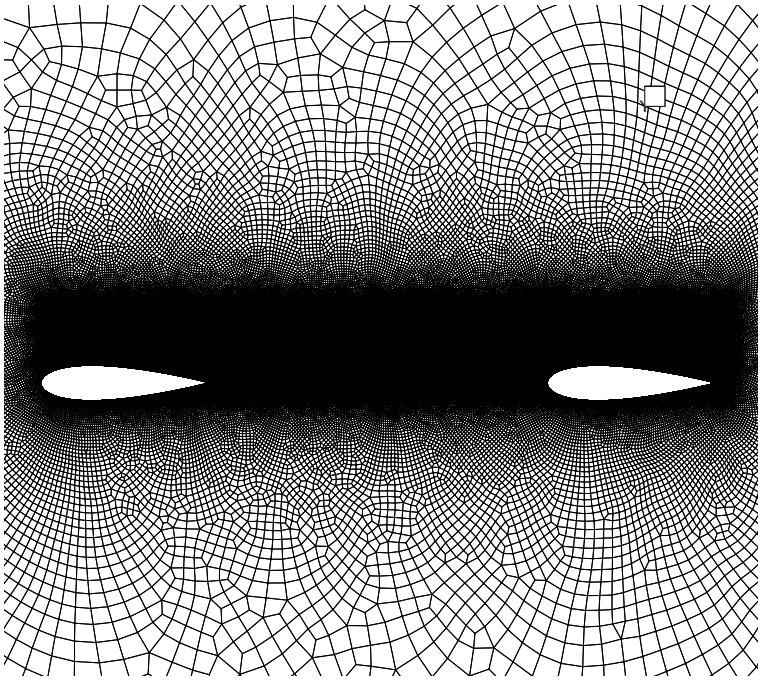
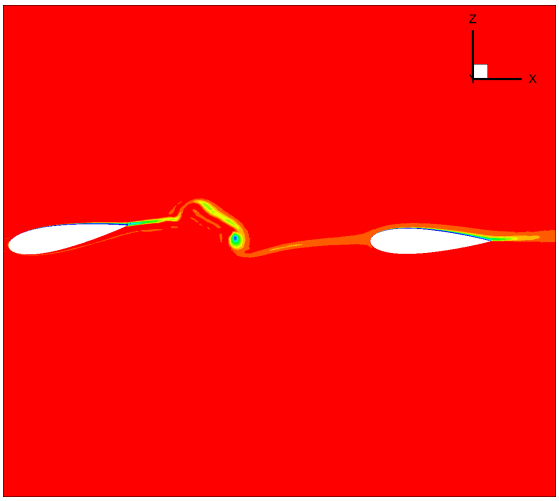
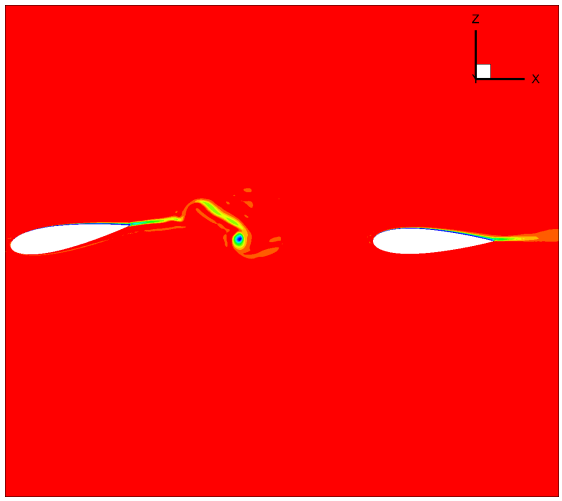


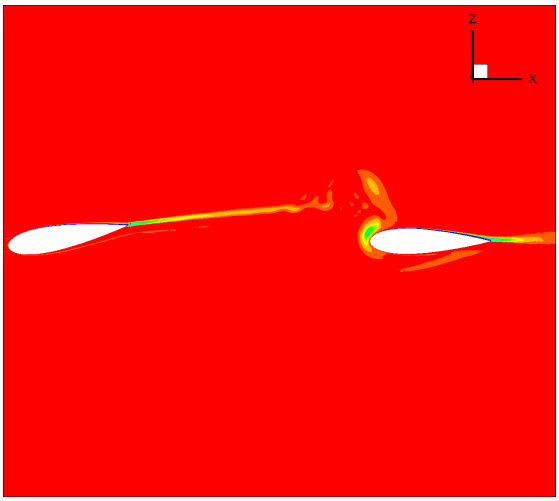
Fig. 14 Fine mesh for tandem airfoil case



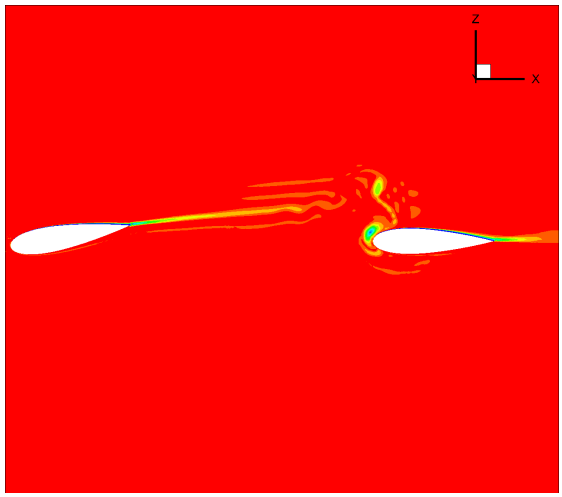
(a) CFD solver on the fine mesh at $t = 2.25 \times 10^{-2}$secs



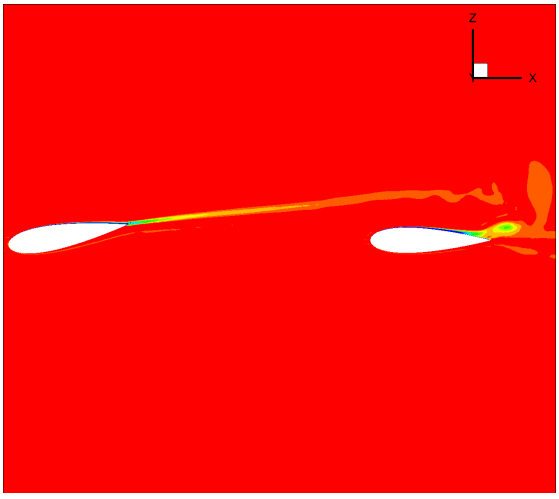
(b) Coupled solver on the coarse mesh at $t = 2.25 \times 10^{-2}$secs



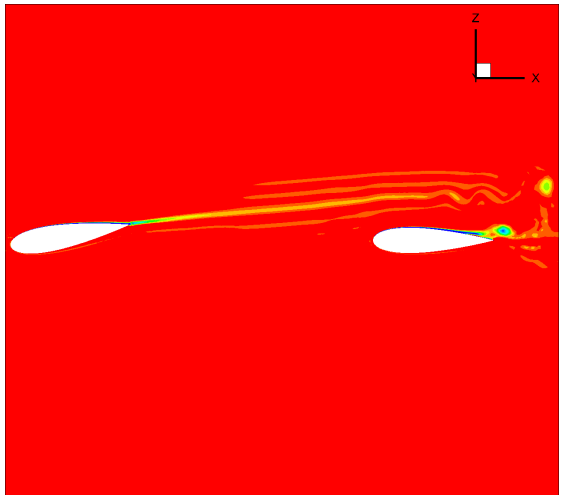
(c) CFD solver on the fine mesh at $t = 4.75 \times 10^{-2}$secs



(d) Coupled solver on the coarse mesh at $t = 4.75 \times 10^{-2}$secs



(e) CFD solver on the fine mesh at $t = 7.25 \times 10^{-2}$secs



(f) Coupled solver on the coarse mesh at $t = 7.25 \times 10^{-2}$secs

Fig. 15 Comparison of vorticity contour for tandem airfoil case

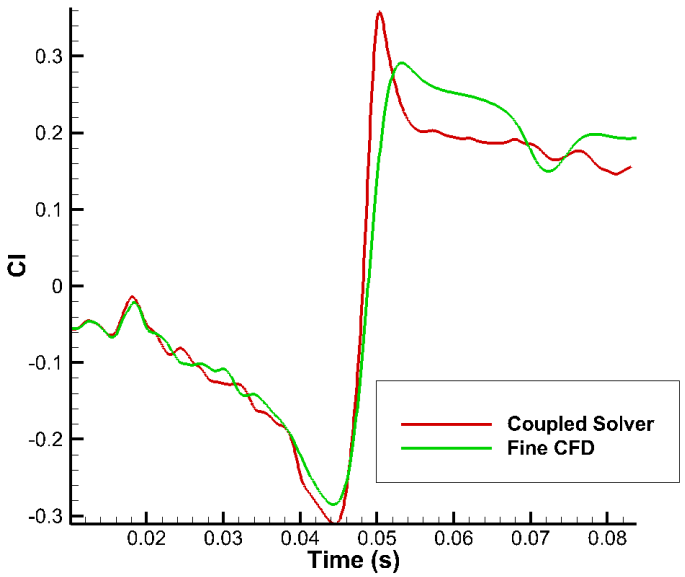


Fig. 16 Comparison of coefficient of lift variation over the trailing aerofoil

References

[1] Srinivasan, G. R., "Computations of Two-Dimensional Airfoil-Vortex Interactions," Tech. Rep. 3885, NASA, 1985.

[2] Potsdam, M., Yeo, H., and Johnson, W., "Rotor Airloads Prediction Using Loose Aerodynamic/Structural Coupling," *Journal of Aircraft*, Vol. 43, No. 3, 2006, pp. 732–742. doi:10.2514/1.14006.

[3] Abate, M., "Gust boundary condition for the DLR TAU-Code," Tech. Rep. IB 124-2007/9, DLR, 2007.

[4] Lin, Y., Savill, M., Vadlamani, R., and Jefferson-Loveday, R., "Wall-Resolved Large Eddy Simulation over NACA0012 Airfoil," *International Journal of Aerospace Sciences*, Vol. 2, No. 4, 2013, pp. 149–162. doi:10.5923/j.aerospace.20130204.01.

[5] Lin, Y., Vadlamani, R., Savill, M., and Tucker, P., "Wall-resolved large eddy simulation for aeroengine aeroacoustic investigation," *The Aeronautical Journal*, Vol. 121, No. 1242, 2017, pp. 1032–1050. doi:10.1017/aer.2017.54.

[6] Lombard, W., Moxey, D., Hoessler, J., Dhandapani, S., Taylor, M., and Sherwin, S., "Implicit Large-Eddy Simulation of a Wingtip Vortex," *AIAA Journal*, Vol. 54, No. 2, 2016, pp. 506–518. doi:10.2514/1.J054181.

[7] Rodriguez, I., Lehmkuhl, O., Borrell, R., and Oliva, A., "Direct numerical simulation of a NACA0012 in full stall," *International Journal of Heat and Fluid Flow*, Vol. 43, 2013, pp. 194–203. doi:10.1016/j.ijheatfluidflow.2013.05.002.

[8] Svard, M., Lundberg, J., and Nordström, J., "A computational study of vortex-airfoil interaction using high-order finite difference methods," *Computers and Fluids*, Vol. 39, No. 8, 2010, pp. 1267–1274. doi:10.1016/j.compfluid.2010.03.009.

[9] Wolf, C., "A Chimera Simulation method and Detatched Eddy Simulation for Vortex-Airfoil Interactions," Ph.D. thesis, University of Göttingen, 2010. URL `http://hdl.handle.net/11858/00-1735-0000-0006-B6A5-2`.

[10] Malan, A., and Oxtoby, O., "An accelerated, fully-coupled, parallel 3D hybrid finite-volume fluid-structure interaction scheme," *Computer Methods in Applied Mechanics and Engineering*, Vol. 253, 2013, pp. 426–438. doi:10.1016/j.cma.2012.09.004.

[11] Cottet, G., and Koumoutsakos, P. D., *Vortex Methods: Theory and Practice*, Cambridge Univeristy Press, 2000.

[12] Rossow, V., "Inviscid modeling of aircraft trailing vortices," Tech. Rep. 409, NASA, 1977.

[13] Smith, S., and Kroo, I., "Induced Drag Computations on Wings with Accurately Model Wakes," *Journal of Aircraft*, Vol. 34, No. 2, 2006, pp. 253–255. doi:10.2514/2.7570.

[14] Chatelain, P., Curioni, A., Bergdorf, M., Rossinelli, D., Andreoni, W., and Koumoutsakos, P., "Billion vortex particle direct numerical simulation of aircraft wakes," *Computer Methods in Applied Mechanics and Engineering*, Vol. 197, No. 13-16, 2008, pp. 1296–1304. doi:10.1016/j.cma.2007.11.016.

[15] Röttgermann, A., Behr, R., Schöttl, C., and Wagner, S., *Numerical Techniques for Boundary Element Methods*, Vieweg+Teubner Verlag, Wiesbaden, 1992, Chaps. Calculation of Blade-Vortex Interaction of Rotary Wings in Incompressible Flow by an Unsteady Vortex-Lattice Method Including Free Wake Analysis, pp. 153–166. doi:10.1007/978-3-663-14005-4_15.

[16] Padakannaya, R., "The Vortex Lattice Method for the Rotor-Vortex Interaction Problem," Tech. Rep. 2421, NASA, 1974.

[17] Sitaraman, J., "CFD based unsteady aerodynamic modeling for rotor aeroelastic analysis," Ph.D. thesis, University of Maryland, 2003. URL `http://hdl.handle.net/1903/310`.

[18] Anusonti-Inthra, P., and Floros, M., "Coupled CFD and Particle Vortex Transport Method: Wing Performance and Wake Validations," *38th Fluid Dynamics Conference and Exhibit*, 2008. doi:10.2514/6.2008-4177.

[19] Zhao, j., He, C., Zhang, L., Zhao, H., and Hu, P., "Coupled Viscous Vortex Particle Method and Unstructured Computational Fluid Dynamics Solver for Rotorcraft Aerodynamic Interaction Analysis," *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2011. doi:10.2514/6.2011-1121.

[20] Stone, C., Duque, E., Hennes, C., and Gharakhani, A., "Rotor wake modeling with a coupled Eulerian and Vortex Particle Method," *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, 2010. doi:10.2514/6.2010-312.

[21] Stock, M., Gharakhani, A., and Stone, C., "Modeling Rotor Wakes with a Hybrid OVERFLOW-Vortex Method on a GPU Cluster," *28th AIAA Applied Aerodynamics Conference*, 2010. doi:10.2514/6.2010-4553.

[22] Palha, A., Manickathan, L., Simao Ferreira, C., and van Bussel, G., "A hybrid Eulerian-Lagrangian flow solver," *ArXiv e-prints*, 2015.

[23] Wales, C., Gaitonde, A., and Jones, D., "Simulation of airfoil gust responses using prescribed velocities," *IFASD, Paris*, 2011.

[24] Huntley, S. J., Jones, D., and Gaitonde, A. L., "2D and 3D gust response using a prescribed velocity method in viscous flows," *46th AIAA Fluid Dynamics Conference*, American Institute of Aeronautics and Astronautics Inc, AIAA, 2016. doi:10.2514/6.2016-4259.

[25] Schwamborn, D., Gerhold, T., and Kessler, R., "The DLR-TAU Code - An Overview," *ONERA-DLR Aerospace Symposium*, 1999.

[26] Speck, R., "Generalized Algebraic Kernels and Multipole Expansions for Massively Parallel Vortex Particle Methods," Ph.D. thesis, Universität Wuppertal, 2011.

[27] Wincklemans, G., and Leonard, A., "Contributions to Vortex Particle Methods for the computation of three-dimensional incompressible unsteady flows," *Journal of Computational Physics*, Vol. 109, No. 2, 1993, pp. 247–273. doi:10.1006/jcph.1993.1216.

[28] Cheng, H., Greengard, L., and Rokhlin, V., "A fast adaptive multipole algorithm in three dimensions," *Journal of Computational Physics*, Vol. 155, No. 2, 1999, pp. 466–498. doi:10.1006/jcph.1999.6355.

[29] Cruz, F., and Barba, L., "Characterization of the errors of the Fast Multipole Method approximation in particle simulations," *International Journal for Numerical Methods in Fluids*, Vol. 79, No. 13, 2009, pp. 1577–1604. doi:10.1002/nme.2611.

[30] Carrier, J., Greengard, L., and Rokhlin, V., "A fast adaptive multipole algorithm for particle simulations," *SIAM Journal on Scientific and Statistical Computing*, Vol. 9, No. 4, 1988, pp. 669–686. doi:10.1137/0909044.

[31] Barba, L., Leonard, A., and Allen, C., "Advances in viscous vortex methods-meshless spatial adaptation based on radial basis function interpolation," *International Journal for Numerical Methods in Fluids*, Vol. 47, No. 5, 2005, pp. 387–421. doi:10.1002/´d.811.

[32] Barba, L., and Rossi, L., "Global field interpolation for particle methods," *Journal of Computational Physics*, Vol. 229, No. 4, 2010, pp. 1292–1310. doi:10.1016/j.jcp.2009.10.031.

[33] Scully, M., "Computation of helicopter rotor wake geometry and its influence on rotor harmonic airloads," Ph.D. thesis, Massachusetts Institute of Technology, 1975. URL `http://hdl.handle.net/1721.1/64826`.

[34] Oh, W. S., Kim, J., and Kwon, O., "Numerical Simulation of Two-Dimensional Blade-Vortex Interactions Using Unstructured Adaptive Meshes," *AIAA Journal*, Vol. 40, No. 3, 2002, pp. 474–480. doi:10.2514/2.1670.

[35] Srinivasan, G., McCroskey, W., and Kutler, P., "Numerical simulation of the interaction of a vortex with stationary airfoil in transonic flow," *22nd Aerospace Sciences Meeting*, 1984. doi:10.2514/6.1984-254.

[36] Tamura, A., Tsutahara, M., Kataoka, T., Aoyama, T., and Yang, C., "Numerical Simulation of Two-Dimensional Blade-Vortex Interactions Using Finite Difference Lattice Boltzmann Method," *AIAA Journal*, Vol. 46, No. 9, 2008, pp. 2235–2247. doi:10.2514/1.30964.